Assignment 8: Introduction to Random Forest Classifier with Python

Shiva Agarwal

May 16, 2024

Assignment Questions

Question 1: Reading and Exploring the Wine Dataset

1. Read the wine dataset stored in from the location `https://gist.github.com/tijptjik/9408623` and store it as a DataFrame.
2. Assign appropriate column names to the dataset using the given attribute names.
3. Print the first five rows of the DataFrame to examine the dataset.
4. Display the statistical insights of the dataset using the `describe()` function.
5. Visualize the distribution of the 'Alcohol' attribute using a histogram.
6. Visualize the distribution of the 'Ash' attribute using a histogram.
7. Visualize the distribution of the 'Class' attribute using a histogram.
8. Create pair plots to visualize the correlations between selected attributes.
9. Generate a correlation matrix and visualize it using a heatmap.

Question 2: Preparing the Data for Model Training

1. Separate the independent variables (features) and the dependent variable (target) into two separate variables, X and y respectively.
2. Split the dataset into training and testing sets with a test size of 33% and a random state of 42.

Question 3: Building and Evaluating the Random Forest Classifier

1. Create a RandomForestClassifier model.
2. Fit the model to the training data.
3. Predict the target variable using the fitted model.
4. Print the accuracy score of the model.
5. Print the confusion matrix to evaluate the performance of the model.

Solution

Question 1: Reading and Exploring the Wine Dataset

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Read the dataset
df = pd.read_csv('/Datasets/RandomForests/wine.data')

# Assign attribute names
attributes = ["Class", "Alcohol", "Malic acid", "Ash", "
    Alcalinity of ash", "Magnesium",
               "Total phenols", "Flavanoids", "
                   Nonflavanoid phenols", "Proanthocyanins"
                   ,
               "Color intensity", "Hue", "OD280/OD315 of
                   diluted wines", "Proline"]
df.columns = attributes

# Print the first few rows of the DataFrame
print("First few rows of the DataFrame:")
print(df.head())

# Display statistical insights
print("\nStatistical insights of the dataset:")
print(df.describe())

# Visualize 'Alcohol' using a histogram
df.hist('Alcohol')
plt.title("Histogram for Alcohol")
plt.show()

# Visualize 'Ash' using a histogram
df.hist('Ash')
plt.title("Histogram for Ash")
plt.show()

# Visualize 'Class' using a histogram
df.hist('Class')
```

```
plt.title("Histogram for Class")
plt.show()

# Create pair plots
df_n = df[["Class", "Alcohol", "Malic acid", "Flavanoids", "Nonflavanoid phenols",
          "Proanthocyanins", "Color intensity", "Hue", "OD280/OD315 of diluted wines",
          "Proline"]]
sns.pairplot(df_n, height=4, kind="reg", markers=".")
plt.show()

# Generate a correlation matrix and visualize it using a heatmap
corr = df.corr()
cmap = sns.diverging_palette(220, 10, as_cmap=True)
sns.heatmap(corr, cmap=cmap, vmax=.3, square=True, linewidths=6, cbar_kws={"shrink": .5})
plt.show()

# Detailed correlation matrix
plt.figure(figsize=(12,12))
plt.title('Pearson Correlation of Features', y=1.05, size=15)
sns.heatmap(df.corr(), linewidths=0.1, vmax=1.0, square=True, cmap=plt.cm.viridis,
            linecolor='white', annot=True)
plt.show()
```

Question 2: Preparing the Data for Model Training

```
# Separate features and target variable
y = df['Class']
X = df.drop(columns=['Class'])

# Split dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
```

Question 3: Building and Evaluating the Random Forest Classifier

```
# Create RandomForestClassifier model
```

```
2  classifier = RandomForestClassifier(n_jobs=2,
       random_state=42)
3
4  # Fit the model to training data
5  classifier.fit(X_train, y_train)
6
7  # Predict the target variable
8  y_pred = classifier.predict(X_test)
9
10 # Print the accuracy score
11 print("Accuracy:", accuracy_score(y_test, y_pred))
12
13 # Print confusion matrix
14 print("\nConfusion Matrix:")
15 print(confusion_matrix(y_test, y_pred))
```